Interconnection optimization for multi-cluster avionics networks

Hamdi Ayed, Ahlem Mifdaoui, Christian Fraboul

http://dx.doi.org/10.1109/ECRTS.2013.25

Interconnection Optimization for Multi-Cluster Avionics Networks

H. AYED [1,2], A. MIFDAOUI[1], C. FRABOUL[2]
[1]University of Toulouse-ISAE, France
[2]University of Toulouse-INP-ENSEEIHT/IRIT, France

***Abstract*—The increasing complexity and heterogeneity of avionics networks make resource optimization a challenging task. In contrast to many previous approaches pursuing the optimization of traffic-source mapping and backbone network analysis, our work presented herein mainly focuses on the optimization of interconnection devices for multi-cluster avionics networks. In this paper, we introduce an optimized interconnection device, integrating novel frame packing strategies and schedulability analysis to enhance the communications between an AFDX-like backbone network and various peripheral sensor/actuator networks in terms of resource savings. The performance analysis conducted on a representative avionics communication architecture highlights the efficiency of our proposal to save resources particularly consumed bandwidth. These latter is considered as an important feature for avionics applications to guarantee easy incremental design during the long lifetime of an aircraft.**

***Keywords*-Interconnection devices, heterogeneous avionics networks, frame packing strategies, schedulability analysis, optimization process**

I. Context and Motivation

The complexity of avionics communication architecture is increasing rapidly due to the growing number of interconnected subsystems and the quantity of exchanged data. To follow this trend, a first avionics architecture was implemented by Airbus in A380, based on a high rate backbone network like the AFDX (Avionics Full Duplex Switched Ethernet) [1] to interconnect critical subsystems. Then, some specific subsystems could have associated sensor/actuator networks based on low rate data buses, such as ARINC429 [2], MIL STD 1553B [3] or CAN [4].

Although this architecture simplifies the design process and reduces the time to market, at the same time it leads in to inherent weight and integration costs due to the large number of sensor/actuator networks. In addition, this architecture makes the avionics subsystems closely dependent on their inputs/outputs and no longer interchangeable. However, for avionics applications, it is essential that the communication architecture fulfills emerging requirements in terms of modularity and performance during the aircraft lifetime.

In order to handle these limitations, the current solution consists of keeping the AFDX as a backbone network to interconnect the critical avionics systems, and dissociating the sensors and actuators from their associated end-systems. As shown in Figure 1, the obtained clusters are interconnected via specific devices, called Remote Data Concentrators (RDCs) and standardized as ARINC655 [5]. RDCs are modular gateways distributed throughout the aircraft to handle heterogeneity between AFDX-based backbone and peripheral data buses.

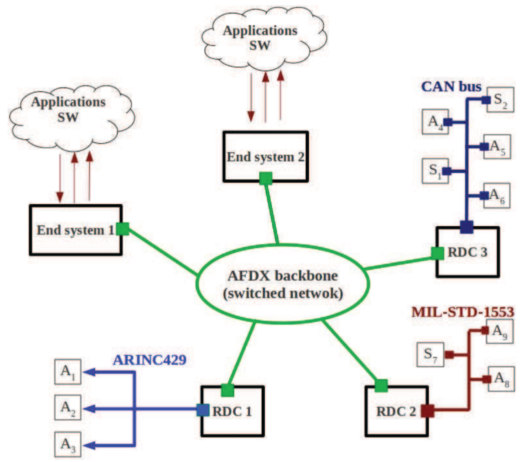

Fig. 1. *Heterogeneous avionics network*

Hence, this alternative architecture enhances the avionics subsystems modularity and simplifies the reconfiguration process. The RDC actually becomes the main node that needs to be reconfigured in case of sensor or actuator modification and at the same time, one of the major challenges in the design process of such multi-cluster avionics networks. The currently implemented RDC in new generation aircraft like the A400M or A350 is based on a naive frame-conversion strategy, called (1:1) strategy, where each non-AFDX frame is converted to an AFDX frame and vice-versa. This strategy is simple to implement, however it implies high network-resource use, and particularly in terms of bandwidth utilization. This feature is inherently important for avionics applications to guarantee easy incremental design and enhance margins for future avionics functional additions. Clearly, resource optimization concepts are necessary to enhance the scalability and performances of avionics applications.

Various resource-optimization solutions have been proposed for critical embedded networks. These approaches range from implementing optimal traffic-source mapping ([6] [7] [8]) to defining optimal routing and analysis algorithms for the network ([6] [9] [10]). However, few researchers have considered the issue of optimizing the interconnection device for multi-cluster embedded networks. In this specific area, some approaches based on the frame packing concept have been proposed [11] [12] [13]. Frame packing consists in building frames from many elementary data packets that respect the maximal frame size in order to reduce the overhead compared to including one data packet in each frame. There are two main classes of frame packing strategies: dynamic and static. The former consists in pooling many data packets in the same frame based on specific criteria. This leads to a dynamic frame structure where the elementary packets can vary from one transmission to another. The latter is based on an offline configuration where elementary data packets pooled in the same frame are explicitly identified a priori. The frame structure is then fixed during transmissions.

In our previous work [13], a novel dynamic frame packing strategy based on a waiting timer, called Fixed Waiting Time (FWT) strategy, was introduced within the RDC device and the results obtained for a representative avionics case study showed an inherent improvement in system performance, compared to a simple (1:1) strategy. In order to obtain further enhancements in terms of resource savings for avionics networks, in this paper we introduce an accurate static frame packing strategy, called Messages-Set Partitioning (MSP) strategy, and we integrate this approach within an optimized RDC device to handle communications between the AFDX-like backbone network and a peripheral sensor/actuator network X.

Hence, our main contributions in this paper are threefold. **First**, the design of an optimized X-AFDX RDC device implementing resource optimization concepts, including dynamic and static frame packing strategies and schedulability analysis, is proposed. **Second**, a concrete implementation of our proposal for a representative avionics application based on AFDX and CAN technologies is specified. **Third**, the ability of this optimized RDC device to improve resource savings and particularly bandwidth consumption is analyzed through a realistic avionics case study.

In the next section, we review the most relevant work in the domain of resource optimization for critical embedded networks and relate them to our work. Afterwards, in section 3, we explain the main concepts of our designed X-AFDX RDC device. These are the frame packing strategies, the generic approach to schedulability analysis and the resource optimization process. In section 4, we present the concrete implementation of this alternative for a representative avionics application, CAN-AFDX, including optimization approaches to find the most effective messages-partitioning in terms of bandwidth consumption. Finally, in section 5, we conduct performance analyses to evaluate the efficiency of our proposal to improve resource savings through a realistic avionics case study.

II. Related work

In the area of optimizing resources over critical embedded networks, various approaches have been proposed and integrated into different parts of the end-to-end communication path, including traffic-sources, communication networks and interconnection devices.

Optimizing traffic-source mapping has been investigated by some researchers and applied to different case studies. For avionics applications, the authors in [6] present an approach to group generated data from different AFDX applications inside the same end-system within the same AFDX frame. This approach enhances end-to-end communication delays and the bandwidth utilization rate on the AFDX. For automotive applications, a similar approach was proposed in [7] in CAN equipment to enhance good throughput on the network. In the context of heterogeneous embedded networks, the authors in [8] presented a design space exploration approach to find optimal software mapping within source nodes, satisfying the different system constraints and minimizing traversal delays.

Various analytical approaches have been used in the literature to optimize the performances offered by critical embedded networks. For avionics applications, many methods have been investigated to analyze the AFDX features and to prove the communication determinism, such as Network Calculus in [14], trajectory approach in [9] and model checking in [15]. Another work proposed in [6] focused on the optimization of routing algorithms for the AFDX network to minimize communication latencies due to load balancing.

In the specific area of optimizing the interconnection device for multi-cluster embedded networks, some approaches based on the dynamic frame packing concept have been proposed. For industrial applications, the authors in [11] present an interesting approach for industrial applications to enhance the CAN scalability using Ethernet to interconnect different CAN buses based on CAN-Ethernet encapsulating bridges. This approach consists in fixing the number of elementary packets in each transmitted frame within the bridges to reduce the induced overhead from CAN bus to Ethernet. Using simulation, obtained communication latencies showed enhanced performances compared to using basic bridges. However, no analytical proof was provided concerning schedulability constraints for hard real-time applications. A second approach proposed in [12] is based on a minimum frame's filling level where data packets are pooled in the same frame until reaching the fixed filling level. This approach could be efficient to reduce the overhead and minimize the bandwidth consumption for non real time applications. However, for real-time communication with hard constraints, this approach can lead to a poor temporal behavior since the schedulability issue was not integrated during the frame packing phase. However, to our knowledge, the implementation of a static frame packing strategy within the interconnection device has not been treated in the literature.

Our proposal in this paper consists in designing an optimized X-AFDX RDC device to interconnect the backbone network AFDX to a peripheral network X and to improve resource savings. This proposed device implements dynamic and static frame packing strategies.

In our previous work [13], a dynamic frame packing strategy, called *Fixed Waiting Time (FWT)* strategy, was proposed. This strategy is based on a waiting timer, as shown in Figure 2. This timer allows the accumulation of many frames at the input

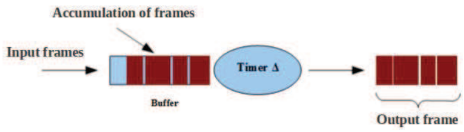

Fig. 2. *Fixed Waiting Time strategy*

interface of the RDC device. Then, when the timer expires, the accumulated data will be sent in the same output frame. This strategy required the adapted choice of waiting time within the RDC device to maximize resource savings. Performance analyses of this strategy have shown an inherent enhancement of resource savings capabilities. However, it is worth noting that these capabilities are limited in case of low input traffic loads.

In order to obtain further enhancements of resource management, we introduce in this paper a new static frame packing strategy, called *Messages-Set Partitioning (MSP)* strategy. Then, these frame packing strategies are combined to the optimization process to maximize resource savings and guarantee schedulability constraints. It is worth noting that the authors in [6] and [7] did not handle these features. The concrete implementation of this proposal is detailed for a representative avionics communication architecture based on AFDX and CAN technologies. Following this, a performance analysis of such an alternative is conducted through a realistic case study to show its efficiency in terms of resource savings.

III. The Design of an Optimized X-AFDX RDC Device

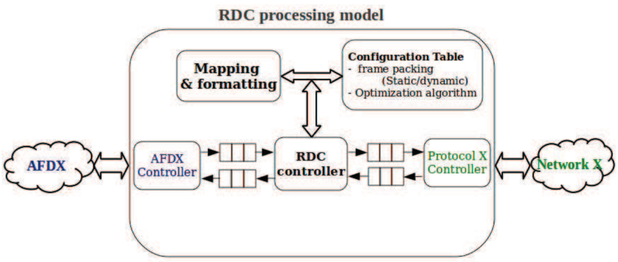

Fig. 3. *Functional model of optimized X-AFDX RDC*

The basic RDC device, implemented currently in avionics communication architecture and following ARINC655 specifications [5], is based on a simple conversion strategy, called (1:1) strategy. This latter proceeds as follows: first, each frame received on the input interface is decapsulated to extract the payload. Then, thanks to the static mapping data-table, the required header is identified and added to the extracted payload and the obtained frame is then sent through the target network interface. This strategy is simple to implement, but it can induce high resource use in both input and output networks. In order to overcome these limitations, our proposal consists of designing an optimized X-AFDX RDC device including: (i) accurate frame packing strategies to reduce communication overheads; (ii) adequate optimization approaches to find the best configuration maximizing resource savings.

The functional model of this proposed device is shown in Figure 3. The configuration table integrates the proposed frame packing strategies and optimization approaches. This configuration impacts the frame mapping and formatting in order to send one or many frames from the input network, on the same frame to the output network. The choice of the frame's structure must take into account network specifications and reduce the induced communication overhead as much as possible. The RDC controller performs these functions to relay the frames from the peripheral network X to the AFDX network and vice-versa. The main arising issues from defining and integrating this optimized RDC device are threefold:

- frame packing strategies: we need to define the strategy used and set its parameters to map the input frames to the output ones;
- schedulability analysis: in order to deal with the worst case performance analysis of such networks and the impact of the introduced RDC device, an appropriate schedulability analysis has to be considered;
- optimization process: this process will define the most accurate configuration respecting the different system constraints and maximizing resource savings.

A. Frame Packing Strategies

The frame packing strategy implemented in the RDC device consists in building a set of output periodic frames $F_{out} = \{i \in [1..N_{out}], f^i_{out}\}$ given a set of input periodic frames $F_{in} = \{j \in [1..N_{in}], f^j_{in}\}$, respecting the protocol characteristic dissimilarities between the input and output networks and minimizing the induced communication overhead.

In order to define the output frames parameters, let's consider for each input frame f^j_{in} a quadruplet $(P^j_{in}, L^j_{in}, Dl^j_{in}, DES^j_{in})$ corresponding respectively to its period, size, deadline and the set of final destinations connected to the output network. Using frame packing strategies, an output frame f^i_{out} containing a subset of input frames $S(f^i_{out}) \subset F_{in}$ is defined by a quadruplet $(P^i_{out}, L^i_{out}, Dl^i_{out}, DES^i_{out})$ where:

- P^i_{out}: represents its production period. Since frame f^i_{out} needs to transmit the message having the smallest period within the subset $S(f^i_{out})$, its period has to respect the following constraint:

$$P^i_{out} \leq \min_{f^j_{in} \in S(f^i_{out})} P^j_{in} \tag{1}$$

- L^i_{out}: is equal to the sum of message payloads in subset $S(f^i_{out})$ and the induced overhead imposed by output network protocol:

$$L^i_{out} = \sum_{f^j_{in} \in S(f^i_{out})} L^j_{in} + overhead \tag{2}$$

- Dl^i_{out}: represents its relative deadline on the output network and depends on the temporal constraints of its associated subset $S(f^i_{out})$;

- DES^i_{out}: represents its set of destinations on the output network. It is simply chosen as the union of destinations of frames in $S(f^i_{out})$:

$$DES^i_{out} = \cup_{f^j_{in} \in S(f^i_{out})} DES^j_{in} \tag{3}$$

For FWT strategy, the subset of input frames $S(f^i_{out}) \subset F_{in}$ is defined on-line and can vary from one transmission to another. However, for MSP strategy, this subset is defined off-line and fixed during transmissions.

The MSP strategy consists in defining off-line input frame partitioning where each sub-partition represents the associated subset of an output frame. The optimized RDC proceeds as shown in Figure 4 when using the MSP strategy. First, the received input frames are queued in the input port of the RDC device. Then, based on a static mapping table, each input frame is relayed to its associated output queue. The frame packing is synchronized with the reception of the most urgent input frame among each defined sub-partition. A timeout could be implemented to avoid losing all the accumulated messages in case of non-reception of the most urgent one. Finally, the output frames will be multiplexed in the output port of the RDC device according to FIFO policy and then transmitted on the output network.

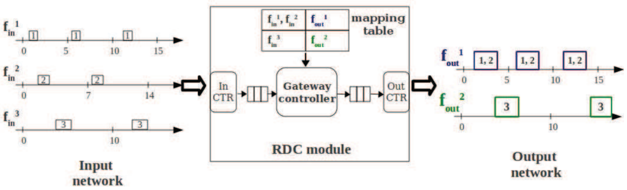

Fig. 4. *MSP packing strategy process*

B. Generic Approach to Schedulability Analysis

For avionics embedded applications, it is essential that the communication network fulfills certification requirements, e.g., predictable behavior under hard real-time constraints and temporal deadline guarantees. The use of a frame packing process within the RDC may increase communication latencies and real-time constraints have to be checked. In order to deal with the worst case performance analysis of such networks, we consider as metric the worst case end-to-end delay that will be compared to the temporal deadline for each frame.

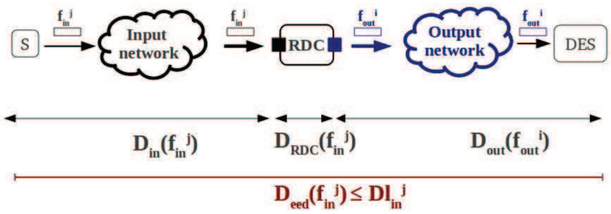

Fig. 5. *End-to-end delay metric definition*

The end-to-end delay for each input frame f^j_{in} in the RDC device, sent in frame f^i_{out} to the output consists of three parts as shown in Figure 5, respectively:

- $D_{in}(f^j_{in})$: a maximal bound on traversal delay of input network for frame f^j_{in};
- $D_{RDC}(f^j_{in})$: the maximal duration the input frame f^j_{in} might be delayed in the RDC device which depends on the selected frame packing strategy;
- $D_{out}(f^i_{out})$: a maximal bound on traversal delay of output network for frame f^i_{out} that includes the payload of input frame f^j_{in};

The system's communication is said to be schedulable if all transmitted frames respect their respective deadlines. The schedulability test is then as follows:

$$\begin{gathered} \forall f^i_{out} \in F_{out} \text{ and } f^j_{in} \in S(f^i_{out}) \subset F_{in}, \\ D_{in}(f^j_{in}) + D_{RDC}(f^j_{in})) + D_{out}(f^i_{out}) \leq Dl^j_{in} \end{gathered} \tag{4}$$

This schedulability test can be written for frame f^i_{out} as follows: $\forall f^i_{out} \in F_{out}$,

$$D_{out}(f^i_{out}) \leq \min_{f^j_{in} \in S(f^i_{out})} \{Dl^j_{in} - (D_{RDC}(f^j_{in}) + D_{in}(f^j_{in}))\} \tag{5}$$

Hence, $\forall f^i_{out} \in F_{out}$,

$$miss(f^i_{out}) = \begin{cases} 0 & \text{(5) is verified} \\ 1 & \text{Otherwise} \end{cases} \tag{6}$$

C. Resource-Optimization Process

In order to increase the efficiency of resource savings on the avionics networks and enhance margins for future function additions, an adequate optimization process is required to define the best RDC configuration maximizing the resource savings and respecting system constraints. Bandwidth consumption is a representative criteria to estimate the resource savings on the network. Thus, our objective consists in finding the best output traffic mapping minimizing bandwidth consumption. The formulation of this optimization problem is as follows.

$$\begin{aligned} \underset{F_{out}}{\text{minimize}} \quad & Bw(F_{out}) = \sum_{f^i_{out} \in F_{out}} \frac{L^i_{out}}{P^i_{out}} \\ \text{subject to} \quad & \forall f^i_{out} \in F_{out}, miss(f^i_{out}) = 0 \end{aligned} \tag{7}$$

where,

- $Bw(F_{out})$ is the sum of reserved bandwidth of output traffic;
- the constraint corresponds to the schedulability of all transmitted frames on the output network.

In [13], this optimization problem was formulated for FWT frame packed strategy and the main parameter to find the optimal solution was the waiting time duration to enhance the efficiency of bandwidth consumption on the output network. The main difficulty in resolving this optimization problem was related to the fact that the end-to-end delays could not be written as a closed form function of the waiting time. Hence, we introduced an optimization approach to find the most accurate value.

For the MSP strategy proposed in this paper, this optimization problem can be modeled as a "Bin Packing" problem, considered like a NP-hard problem in [16], where output frames are considered as the bins and input frames are the objects to put into these bins. In ”Bin packing”, the number of used bins has to be minimized, this corresponds in our case to minimizing induced bandwidth consumption.

Many optimization approaches can be used to find a feasible frame partition that respects the schedulability constraints and minimizes induced bandwidth on networks. Three main approaches can be cited. The first one is based on Exhaustive Search that considers all possible partitions to find a feasible and optimal solution. However, the partition number of a set with size n is known as Bell number B that grows exponentially with n, i.e., for a set of 20 frames, $B \sim 5.10^{14}$. Hence, this approach will certainly lead to the best frame partition in terms of bandwidth consumption, but at the same time it is a time-consuming approach due to solutions-space explosion. The second approach consists in building a specific heuristic for our problem to have a feasible and acceptable solution in terms of bandwidth consumption within a polynomial time. The third approach is based on the optimal algorithm Branch & Bound to bridge the gap between the Exhaustive Search and the Heuristic approach by reducing the size of explored solutions-space compared to the former and enhancing the quality of the obtained solution compared to the latter.

TABLE I
COMPARATIVE ANALYSIS OF OPTIMIZATION APPROACHES

Approach	Complexity	Solution Quality
Exhaustive Search	high	high
Branch & Bound	medium	high
Heuristic	low	medium

A comparative analysis between the three approaches in terms of complexity and solution quality is shown in Table I. Hence, as can be noticed, the Heuristic approach and the Branch & Bound algorithm are the most adapted to our optimization problem.

IV. CAN-AFDX APPLICATION

In this section, the generic concepts presented in section III are illustrated for a realistic avionics communication architecture based on AFDX technology for the backbone network and CAN technology for sensor/actuator networks. First, the basic concepts of these technologies are presented. Then, the configuration of the proposed frame packing strategy MSP is detailed. Finally, schedulability analysis and optimization process are developed.

A. Network Technologies

The AFDX [1] network is based on the Full Duplex Switched Ethernet protocol at 100Mbps. This technology manages the large amount of exchanged data through policing mechanisms added in switches and the Virtual Link (VL) concept. This concept provides a way to reserve a guaranteed bandwidth for each traffic flow. The VL represents a multicast communication which originates at a single End System and delivers packets to a fixed set of End Systems. Each VL is characterized by: (i) BAG (Bandwidth Allocation Gap), ranging in powers of 2 from 1 to 128 milliseconds, which represents the minimal inter-arrival time between two consecutive frames; (ii) MFS (Maximal Frame Size), ranging from 64 to 1518 bytes, which represents the size of the largest frame that can be sent during each BAG.

CAN native protocol [4] is a 1 Mbps data bus that operates according to an event-triggered paradigm where messages are transmitted using a priority-based access mechanism. Collisions on the bus are resolved following a CSMA/CR protocol (Carrier Sense Multiple Access/Collision Resolution) thanks to the bit arbitration method. CAN frame includes a payload up to 8 bytes and an overhead of 6 bytes due to the different headers and bit stuffing mechanism.

CAN-AFDX RDC device needs to handle the dissimilarities between CAN and AFDX in terms of communication paradigms and protocol characteristics. The frame size difference between these two networks shows that the frame packing in the RDC device is required for the communication direction from CAN to AFDX to maximize the resource savings, and not along AFDX to CAN. Hence, in this part, we will specify the RDC function for the communication direction from CAN to AFDX to illustrate a concrete configuration of the MSP strategy and the schedulability and optimization process.

B. Frame packing strategy

For an avionics communication architecture based on AFDX and CAN technologies, the application of MSP strategy in the CAN-AFDX RDC, considering CAN to AFDX direction, consists in building a set of AFDX VLs $V = \{v_1, v_2, ..., v_m\}$ to define the output traffic from the RDC to the AFDX network, given a set of CAN-messages $M = \{m_1, m_2, ..., m_n\}$ at the input. This partition has to be schedulable i.e., all CAN messages in M respect their deadlines; and the allocated VLs V minimize the induced bandwidth rate from the RDC to the AFDX.

For each message $m_j \in M$, we associate four characteristics $\{P^j, L^j, D^j, DES^j\}$ which represent the period, maximum payload, deadline and the set of AFDX destinations respectively.

Each AFDX frame within each VL $v_i \in V$, obtained after the frame packing, will contain a static subset of CAN messages $M(v_i) \subset M$ that does not change over successive transmissions. Each VL v_i is characterized by $\{BAG^i, MFS^i, Dl^i, DES^i_{AFDX}\}$ where:

- BAG^i: corresponds to P^i_{out} in the general case; since each allocated VL v_i in the RDC needs to transmit the message with the smallest period within the subset $M(v_i)$ coming from the associated CAN bus to the AFDX, we define the BAG as the closest value of power of 2 to the smallest period of messages in $M(v_i)$:

$$BAG^i = 2^k, \quad k = \left\lfloor \frac{log(min_{m_j \in M(v_i)} P^j)}{log(2)} \right\rfloor \tag{8}$$

- MFS^i: corresponds to L^i_{out} in the general case; it is consequently the sum of the respective payloads of messages in subset $M(v_i)$ and the induced overhead imposed by the AFDX structure (at most 67 bytes); the padding is used to guarantee a minimum AFDX frame size of 84 bytes (IFG (Inter Frame Gap) included):

$$MFS^i = \max\left(84, \sum_{m_j \in M(v_i)} L^j + 67\right) \tag{9}$$

- Dl^i: is the relative deadline of the obtained AFDX frame which depends on its associated CAN-messages subset $M(v_i)$.
- DES^i_{AFDX}: corresponds to DES^i_{out} in the general case; it is the union of destinations of CAN-messages in $M(v_i)$ where $DES^i_{AFDX} = \cup_{m_j \in M(v_i)} DES^j$.

C. Schedulability analysis

The end-to-end delay of each CAN message $m_j \in M(v_i)$, where $M(v_i)$ is the static subset of CAN-messages associated with the VL $v_i \in V$, consists of three parts:

- $d_{CAN}(m_j)$: the maximal response time of a CAN frame. The schedulability analysis for a native CAN bus, considered in this paper, has been considered in [17], where the CAN bus is modeled as a non-preemptive Rate Monotonic scheduler. In our case, the tool *Cheddar* [18] is used to compute this bound;
- $d_{RDC}(m_j)$: the maximal duration the message might be delayed in the RDC. This delay is the sum of: (i) a technological latency due to payload extraction and relaying process, called ϵ; (ii) waiting time in the RDC between the reception instant of the CAN message and the transmission instant of its associated AFDX frame, called $WT(m_j)$, then

$$d_{RDC}(m_j) = \epsilon + WT(m_j) \tag{10}$$

The worst case waiting time of a CAN-message $m_j \in M(v_i) \backslash \{m_s\}$, where m_s is the message with the smallest period, occurs when it arrives immediately after the end of m_s reception in the RDC. In this case, the message m_j has to wait for the next reception of m_s to be packed in the same AFDX frame, as illustrated in Figure 6. Therefore, an upper bound of the waiting time in the RDC of $m_j \in M(v_i)$ is:

$$WT(m_j) \quad = \quad \begin{cases} 0 \ \ if \ \ j = s \\ T_s + d_{CAN}(m_s) \ \ otherwise \end{cases} \tag{11}$$

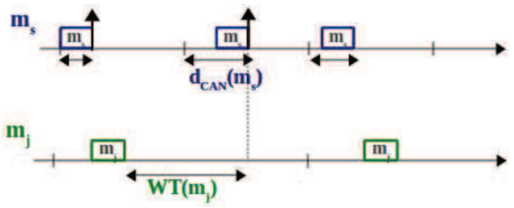

Fig. 6. *Worst case waiting time*

- $d_{AFDX}(v_i)$: the upper bound on the delay submitted by the AFDX VL including m_j. Schedulability analysis for an AFDX network, based on *Network Calculus* formalism, has been introduced in recent work [14]. The tool WoPANets [19] will be used herein to analyze this delay bound.

The schedulability test is: $\forall v_i \in V$:

$$\begin{aligned} d_{AFDX}(v_i) \quad &\leq \quad \min_{m_j \in M(v_i)} \{D^j - (d_{RDC}(m_j) + d_{CAN}(m_j))\} \\ &\leq \quad Dl^i \end{aligned} \tag{12}$$

Hence, the schedulability test becomes: $\forall v_i \in V$,

$$miss(v_i) = \begin{cases} 0 & \text{(12) is verified} \\ 1 & \text{Otherwise} \end{cases} \tag{13}$$

D. Optimization process

Since the MSP strategy configuration is considered as an NP-hard problem, an adapted heuristic approach, called *Bandwidth-Best-Fit Decreasing* (BBFD) heuristic, and an adequate algorithm based on the *Branch & Bound* (B&B) concept are introduced to find a CAN-messages partition, respecting the schedulability constraints and minimizing bandwidth consumption.

a) Bandwidth-Best-Fit Decreasing (BBFD) Heuristic: Several heuristics were introduced to compute an approximate solution for the classical Bin Packing problem [16]. The simplest heuristic is *First-Fit Decreasing (FFD)* based on sorting objects according to a decreasing order of sizes and then inserting them in the first suitable bin. A more effective heuristic is *Best-Fit Decreasing (BFD)* which differs from the first one by selecting the most suitable bin instead of the first suitable one. Our objective is minimizing bandwidth consumption instead of the number of used frames (bins in general case) and guaranteeing the temporal constraints of all the transmitted frames. Thus, we introduce the Bandwidth-Best-Fit Decreasing (BBFD) heuristic described in the flow chart of Figure 7:

(1) Initialization

The CAN-messages with the same period are grouped together to form a messages-class. For each period T_i, we associate a messages-class c_i. In order to reduce the problem complexity, the CAN-messages in the same class are packed together in the same AFDX VL, with respect to its maximal MFS size of 1518 bytes. For a messages-class, we define a deadline as the smallest deadline among its composing messages. Then, the heuristic sorts the messages-classes in increasing order of their respective deadlines. The heuristic starts packing messages with small deadlines first to build partitions favoring the most constrained messages.

(2) Iterative partitioning

The set of AFDX VLs is built iteratively. At the beginning, the first messages-class in the set C is inserted in a new VL that would be added to the set V. Then, the *Deadline-Best-Fit Increasing* heuristic is conducted for each selected messages-class $c_i \in C$ as follows:

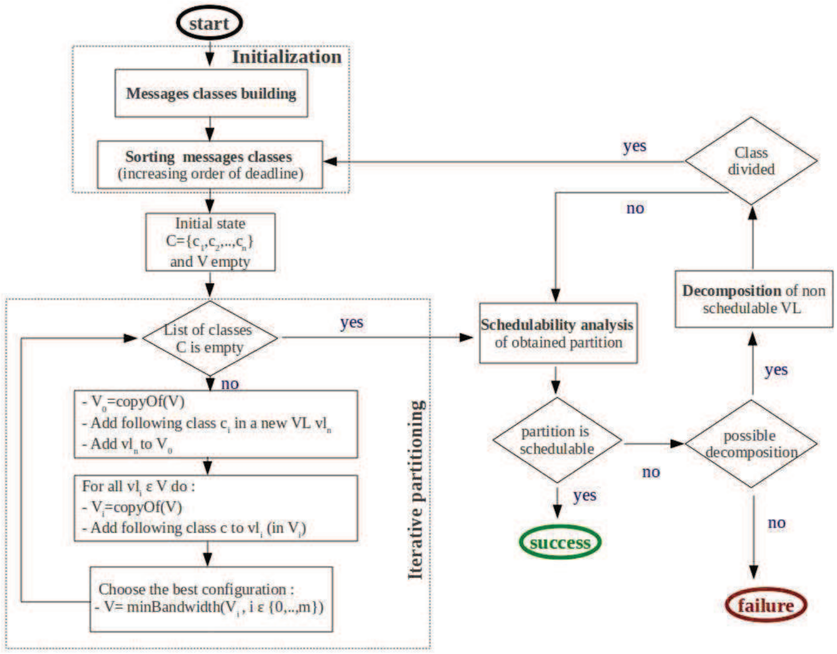

Fig. 7. *Bandwidth-Best-Fit Decreasing heuristic*

- (a) if there is at least one existent VL in V that can support the class c_i, i.e. the associated miss-value equal to 0 according to (13) and the maximal MFS size of 1518 bytes is respected, then it builds the subset $V(c_i)$ that corresponds to the obtained VLs including c_i. Afterward, it selects the VL in $V(c_i)$ that minimizes the obtained bandwidth consumption and adds it to the set V;
- (b) if there is no existent VL in V that can support c_i, because the maximal MFS size of 1518 bytes is exceeded or its associated miss-value equal to 1 according to (13), then it builds a new VL including c_i and adds it to the set V.

At the end of this step, an AFDX VLs set V is obtained such that bandwidth consumption is minimized. However, the schedulability analysis of this configuration needs to be proved.

(3) Schedulability analysis

This step consists in conducting the schedulability analysis explained in section IV-C where the AFDX delay of each VL in the obtained set V needs to be calculated. If the schedulability test in equation (12) is verified, then this configuration is considered as the solution for our optimization process and the heuristic stops successfully. Otherwise, a decomposition process is launched.

(4) Decomposition

The main idea of decomposition consists in identifying the VLs subset $V^* \subset V$, that does not verify the schedulability test in eq. (12). Then, in order to relax this constraint, the heuristic is based on unpacking the most urgent messages-classes included in the identified VLs. Therefore, for each VL $v_k \in V^*$:

- (a) if v_k is composed of only one messages-class, then we split the associated CAN-messages set into two equal subsets in terms of messages-number, if possible, and go back to the step (1) of the heuristic;
- (b) if v_k contains at least two messages-classes, then unpack the most critical class and include it in a new VL. Afterward, update the VL set V and go back to step (3) of the heuristic;
- (c) if v_k consists of one CAN-message, then there is no feasible solution and the heuristic stops and indicates a failure.

b) Branch & Bound Algorithm: The main idea of Branch &Bound algorithm [20] is based on the definition of upper and lower bounds for an objective function in order to explore the most promising subspace of potential solutions, and consequently to reduce the computation's complexity. The two main operations to process the Branch & Bound algorithm are: (i) a Branching-Strategy that consists in generating the new states from an existing one; (ii) a Discarding-Policy that consists in eliminating the subspace of solutions admitting their lower bound of the objective function exceeding the upper bound of a reference solution.

This general algorithm is adapted to our CAN-messages set partitioning problem to build a schedulable set of AFDX VLs within the gateway minimizing the bandwidth consumption. In order to explore the subspace of potential solutions, we identify each state by $(V, \mathcal{C})$, where V is the set of VLs already created and $\mathcal{C}$ the set of messages-classes not yet included in existing VLs in V. The upper and lower bounds of our objective function i.e. bandwidth consumption, the Branching-Strategy and Discarding-Policy are defined for our problem.

(1) Upper bound

The main idea consists in enhancing the quality of the solution obtained with the BBFD heuristic. Hence, the CAN-messages partition obtained with this latter is considered as a reference solution and the induced bandwidth consumption of the obtained VLs set is identified as the upper bound of the bandwidth consumption to launch the exploration of the most promising solutions. If BBFD fails to find a feasible solution, then any schedulable partition of messages-classes can be considered as the reference point, such as partition obtained with (1:1) strategy. However, it is worth noting that the better is the reference solution, the faster an optimal solution is obtained from the B &B algorithm. Each time we find a CAN-messages partition with a bandwidth consumption lower than this upper bound, the reference solution is updated.

(2) Lower bounding function

For each state s characterized by V and $\mathcal{C}$, a lower bound on the bandwidth consumed is defined as follows:

$$lowerBound(s) = Bw(V) + Bw(\mathcal{C}) \tag{14}$$

where, $Bw(\mathcal{C}) = \sum_{c_i \in \mathcal{C}} \sum_{j \epsilon c_i} \frac{L^j}{T^i}$

(3) Branching-Strategy

For each state, we consider all possible states that can be obtained by selecting a messages-class c_i from $\mathcal{C}$ and packing it in an existing VL in V, or by creating a new VL including only c_i.

(4) Discarding-Policy

Three main conditions must be verified in our case to discard a state or to keep it in the potential solutions space. The first concerns the validity of the state where all the generated VLs have miss-value equal to 0 according to (13) and respect the maximal length of 1518 Bytes. The second is related to the fact that the obtained lower bound of the objective function of the state explained in equation (14) has to be smaller than the reference's upper bound. Finally, the third concerns the schedulability of the state which is applicable only for final states that define a complete CAN-messages partition.

The different steps of this optimization method are as follows: first, CAN-messages with the same period are grouped together to form a messages-class with respect to the maximal size of the AFDX frame. Then, a reference solution is obtained using the BBFD heuristic. Afterwards, we apply iteratively the Branching-Strategy and Discarding-Policy: if a state corresponds to a complete partition, i.e. all messages-classes are assigned to AFDX VLs, we proceed to an update of the reference solution only if it enhances the bandwidth consumption and is schedulable, otherwise this state is discarded. If the state is intermediary, which means it corresponds to a partial partition of messages-classes set, then we generate all possible new states by including the messages-class in an existing VL or creating a new VL. For each valid created state, we evaluate the lower bounding function and this state is added to the list to explore only if its lower bound is smaller than the bandwidth consumption of the reference solution. The set of states to explore is sorted in increasing order of lower bounding function value in order to consider the most promising states first.

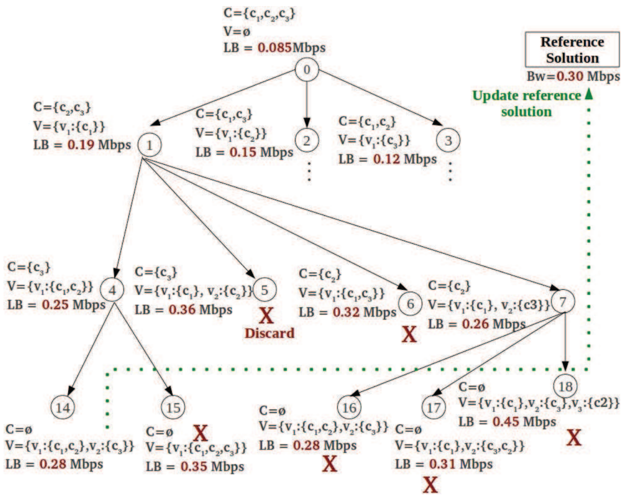

Fig. 8. *B&B based algorithm: an example of traffic with 3 messages-classes*

In Figure 8, the approach is applied to a frame packing example with 3 messages-classes. Only part of the exploration tree is presented in this figure to illustrate its implementation.

V. Performance Evaluation

In this section, we conduct a performance analysis to evaluate the efficiency of our proposed optimized RDC device to improve resource savings through a realistic avionics case study. The impact of optimization approaches and frame packing strategies is illustrated herein.

A. Case study

Our case study is a realistic avionics communication architecture based on a backbone network AFDX and several peripheral CAN sensor buses, as shown in Figure 9.

The traffic circulating on the AFDX Backbone, excluding the traffic generated by sensor networks, is described in

Table II and called "background AFDX traffic" in the rest of this paper. As can be seen, this background AFDX traffic is composed of 450 VLs with BAGs in $\{4, 16, 32\}$ms and MFSs in $\{16, 226, 482\}$ bytes. This table represents the VL distributions according to BAGs and frame lengths values.

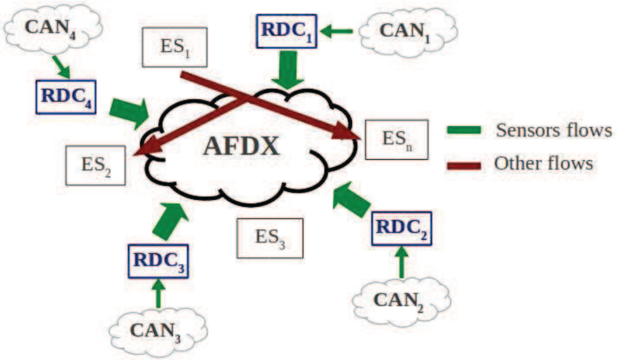

Fig. 9. *Avionics communication architecture based on CAN and AFDX technologies*

TABLE II
THE BACKGROUND AFDX TRAFFIC DESCRIPTION

BAG(ms)	Number of VLs	MFS (bytes)	Number of VLs
4	62	16	386
16	100	226	56
32	288	482	8

TABLE III
CONSIDERED TEST CASES

Test case	Number of buses	% of max CAN load
Test case 1	1 bus	[1..66]
Test case 2	4 buses	[1..40] for each bus

In addition to this background AFDX traffic, we consider CAN sensor traffic generated according to two test cases, described in Table III. The first one consists in varying the traffic load percentage from 1 to 69 for one CAN bus, whereas the second consists in varying the traffic load percentage from 1 to 40 on each CAN bus in case of four buses. The limitation of the traffic load in the second test case (only 40% on each CAN bus) is necessary to guarantee the AFDX stability.

The aim of these analyses is to show the impact of the proposed optimized RDC device on the system performance. First, we consider the test case 1 to show the impact of the optimization approach combined with the MSP strategy on resource savings. Then, the two test cases are used to show the impact of the frame packing strategies on the system's performances.

B. *Obtained Results*

Impact of optimization approaches

In order to analyze the impact of the optimization approach on the MSP strategy performances in terms of maximizing resource savings, a comparative study between the solutions obtained with the two introduced optimization approaches, BBFD heuristic and B&B algorithm, is conducted based on test case 1.

The number of explored states with each approach are described in Table IV to show their respective complexities with reference to Exhaustive Search (ES) approach. Only the scenarios leading to the best enhancements in terms of bandwidth consumption are presented when applying *B&B* algorithm compared to *BBFD* heuristic.

TABLE IV
IMPACT OF OPTIMIZATION APPROACHES ON BANDWIDTH CONSUMPTION

Messages-classes number	2	3	4	5	6	7
States (Heuristic)	3	5	8	11	15	19
States (B&B)	6	45	340	4110	67165	$\geq 1.6\ 10^6$
States (ES)	6	45	508	8285	190000	$\geq 5.6\ 10^6$
$\frac{Bw(BB)-Bw(H)}{Bw(BB)}(\%)$	0	0.2	0.7	0.3	0.1	0.1

As can be seen, the enhancements obtained in terms of bandwidth consumption when applying the *B&B* algorithm instead of the *BBFD* heuristic approach are negligible (less than 1%), whereas the number of explored states with the former is inherently higher compared to the number obtained with the latter. Therefore, our introduced heuristic approach is considered as an accurate approach to find a valid solution with low computing complexity. The *BBFD* heuristic is then selected to find a configuration of the MSP strategy within the RDC and the following performances analyses are based on this approach.

Impact of frame packing strategies

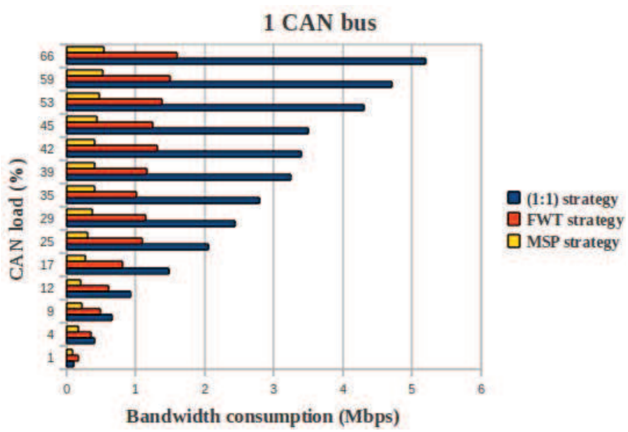

Fig. 10. Impact of frame packing strategies on bandwidth consumption for test case 1

In order to analyze the efficiency of the MSP strategy, we consider the two test cases 1 and 2. The obtained bandwidth rates are illustrated in Figures 10 and 11, respectively. As can be seen, the basic strategy (1:1) leads to an inherent induced bandwidth rate. This is essentially due to the overhead when sending each sensor message (less than 8 bytes) in one AFDX frame (at least 64 bytes). On the other hand, the FWT strategy offers a notable amelioration on the induced bandwidth rate in the AFDX where a reduction of 50% is obtained compared to the (1:1) strategy.

However, the obtained value with the MSP strategy introduced in this paper shows an inherent enhancement compared with the two first strategies where amelioration factors of 5 and 3 are noticed with reference to the (1:1) and FWT strategies, respectively. This is mainly due to the overhead reduction under the MSP strategy by defining explicitly the CAN messages-set packed in each AFDX frame transmitted by the RDC. This clearly leads to an accurate VL allocation induced by the RDC and avoids the over-dimensioning problem.

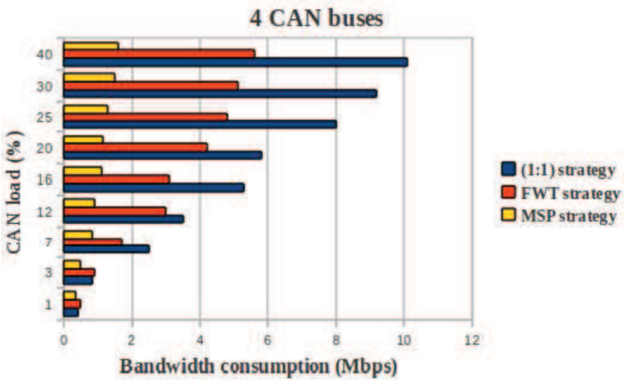

Fig. 11. Impact of frame packing strategies on bandwidth consumption for test case 2

Hence, the MSP frame packing strategy integrated in the CAN-AFDX RDC offers an inherent bandwidth savings on the AFDX, compared to the FWT strategy. These results have shown the efficiency of the proposed RDC device, including frame packing strategies and the optimization process, to save resources and particularly consumed bandwidth for multi-cluster avionics networks.

VI. Conclusion and Future Work

Since efficient resource management is inherently important to guarantee an easy incremental design process for avionics applications, an optimized X-AFDX RDC device is introduced in this paper to interconnect an AFDX backbone to a peripheral network X.

The main issues to define this optimized RDC are: (i) the selection and configuration of the frame packing strategy; (ii) the schedulability analysis to guarantee system constraints; (iii) the resource optimization process to find the best configuration in terms of resource savings. The illustration of these concepts was illustrated for a representative avionics communication architecture based on AFDX and CAN technologies. We first introduced a new frame packing strategy in the RDC device, called *Messages Set Partitioning* (MSP). Then, adequate optimization approaches were detailed, including the BBFD heuristic and an adaptation of the Branch & Bound algorithm, to maximize resource savings with this new frame packing strategy. Finally, the performance analysis of our proposal highlights its capabilities to improve resource savings on avionics networks, and particularly the use of the MSP strategy based on the BBFD heuristic.

The next step in our work consists in analyzing the adaptability of the proposed concepts to the specificities of other sensor/actuator networks like MIL-STD-1553 and TTP/C. Another issue concerns the analysis of other metrics for resource optimization, such as energy savings and minimizing delays, considered as important features for avionics applications.

References

[1] Airlines Electronic Engineering Committee, "Aircraft Data Network Part 7, AFDX NETWORK, ARINC Specification 664." Annapolis, Maryland: Aeronautical Radio, 2002.

[2] Avionic Systems Standardisation Committee, "Guide to avionics data buses," 1995.

[3] Condor Engineering Incorporated, "MIL-STD-1553 Designer guide," 1982. [Online]. Available: http://www.condoreng.com/support/downloads/tutorials/MIL-STD-1553Tutorial

[4] Robert Bosch GmbH, "CAN specification Version 2,0," 1991.

[5] Airlines Electronic Engineering Committee, "Remote Data Concentrator(RDC) generic description." Annapolis, Maryland: Aeronautical Radio, 1999.

[6] A. A. Sheikh, O. Brun, M. Chramy, and P. Hladik, "Optimal Design of Virtual Links in AFDX Networks," *Laas*, 2012.

[7] R. Saket and N. Navet, "Frame packing algorithms for automotive applications," *Journal of Embedded Computing*, vol. 2, pp. 93–102, 2006.

[8] P. Pop, P. Eles, and Z. Peng, "Schedulability-driven frame packing for multicluster distributed embedded systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 4, February 2005.

[9] H. Bauer, J.-L. Scharbarg, and C. Fraboul, "Improving the worst-case delay analysis of an AFDX network using an optimized trajectory approach," *IEEE Transactions on Industrial Informatics, IEEE, Vol. 6 N. 4, p. 521-533*, 2010.

[10] A. Mifdaoui, F. Frances, and C. Fraboul, "Performance analysis of a Master/Slave switched Ethernet for military embedded applications." *IEEE Transactions on Industrial Informatics*, vol. 6, November 2010.

[11] J. Scharbarg, M. Boyer, and C. Fraboul, "CAN-Ethernet architectures for real-time applications." IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2005.

[12] P. Narasimhan, L. Moser, and P. Melliar-Smith, "Message packing as a performance enhancement strategy with application to the totem protocols." In Global Telecommunications Conference (GLOBECOM), Communications, 1996.

[13] H. Ayed, A. Mifdaoui, and C. Fraboul, "Frame Packing Strategy within Gateways for Multi-cluster Avionics Embedded Networks," *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2012.

[14] J. Grieu, "Analyse et valuation de techniques de commutation ethernet pour l'interconnexion de systemes avioniques," Ph.D. dissertation, INP, Toulouse, 2004.

[15] H. Charara, J. Scharbarg, J. Ermont, and C. Fraboul, "Methods for bounding end-to end delays on an AFDX network." Dresden, Germany: Proceedings of the 18th Euromicro Conference o n Real-Time Systems (ECRTS06), 2006.

[16] M. R. G. E. G. Coffman and D. S. Johnson, *Approximation algorithms for NP-hard problems*. PWS Publishing Company, 1996.

[17] R. Davis, A. Burns, R. Bril, and J. Lukkien, "CAN schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, 2007.

[18] "Cheddar project home page." [Online]. Available: http://beru.univ-brest.fr/ singhoff/cheddar/index-fr.html

[19] A. Mifdaoui and H. Ayed, "WOPANets: a tool for WOrst case Performance Analysis of embedded Networks," *CAMAD 2010*, 2010.

[20] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations Research*, 1966.